

Réalité augmentée

Marqueur Image

Christophe Vestri

Le mardi 4 février 2020

Plan du cours

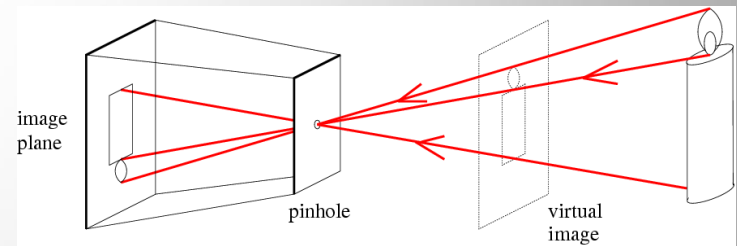
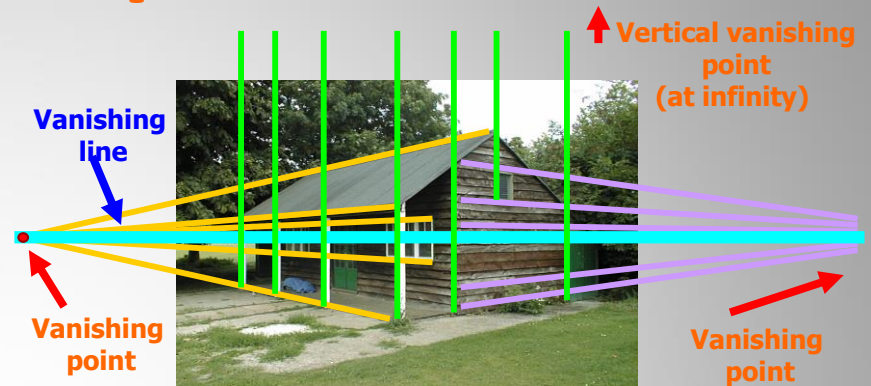
- 7 janvier : Réalité augmentée intro, Unity/Vuforia et projet
 - 15 janvier: Construction application RA, Unity StarWars
 - 28 janvier: Vision par ordinateur (1) et Unity Roll-a-ball
 - 4 février: Vision par ordinateur (2) et ARFondation-Wikitude + projet
 - 3 mars : QRCode, Résumé et présentation des Projets
-
- **Suite: Cours Cartographie/JS/AR/VR**

Plan Cours 4

- Rappel
- Vision par ordinateur 2
- Présentation outils
- Projet Final - finalisation

Rappel du cours précédent

- Points et droites de fuite
- Modèle de caméra
Pinhole et matrice de projection
- Coordonnées homogènes

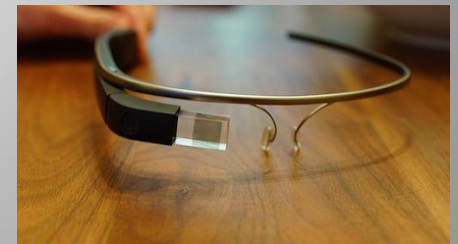


$$\mathbf{x} = \mathbf{K} \begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix} \mathbf{X}$$

$$(x, y) \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

RA avec caméra Mobile

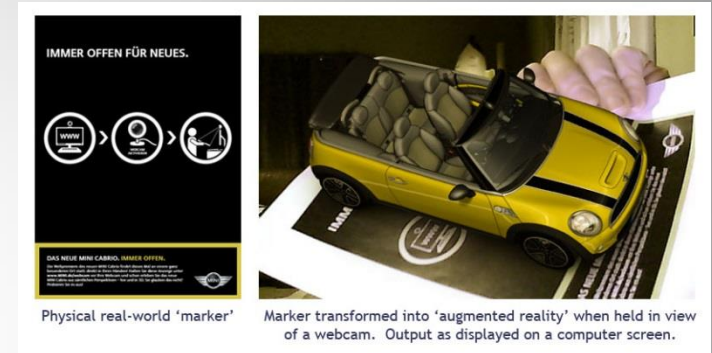
- Smartphones, tout pour la RA
 - Camera + écran – déterminer/montrer ce qui doit être vu
 - Donnée GPS– localisation
 - Compas – quelle direction on regarde
 - Accéléromètre – orientation
 - Connection Internet – fournir des données utiles
- 58% des Français ont un smartphone en 2015
- 90% des 18-24ans
- Lunettes de RA et VR



Types de RA mobile

Marqueurs:

- Caméra pour détecter un marqueur dans le monde réel
- Calcul de sa position et orientation
- Augmente la réalité



Géolocalisation:

- GPS pour localiser son téléphone
- Recherche de Point d'intérêt proche de nous
- Mesure orientation (compas, accéléromètre)
- Augmente la réalité



Types de RA mobile

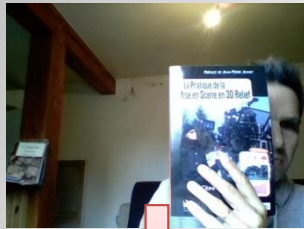
Utilisation de marqueurs caméras:

- Marqueurs Spécifiques:
 - Tag visuels
 - Formes spécifiques (carrés, cercles)
- Marqueurs Images
 - Photo, image de l'objet/scène
- Processus de RA
 - Détection du marqueur dans la vidéo
 - Transformation 2D-3D
 - Affichage 3D



Technologies nécessaires

Références Acquisition vidéo



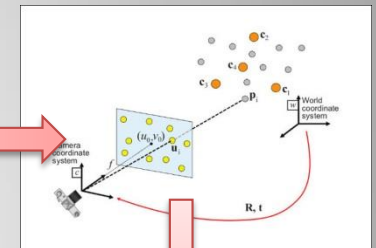
Détection coins et descripteurs



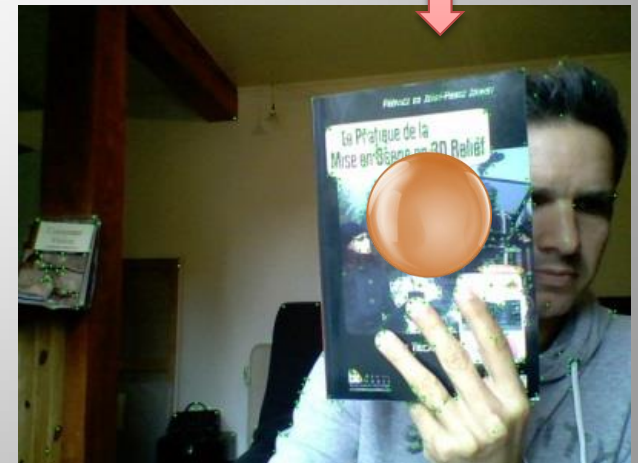
Matching



Calcul de la pose caméra



Affichage 3D

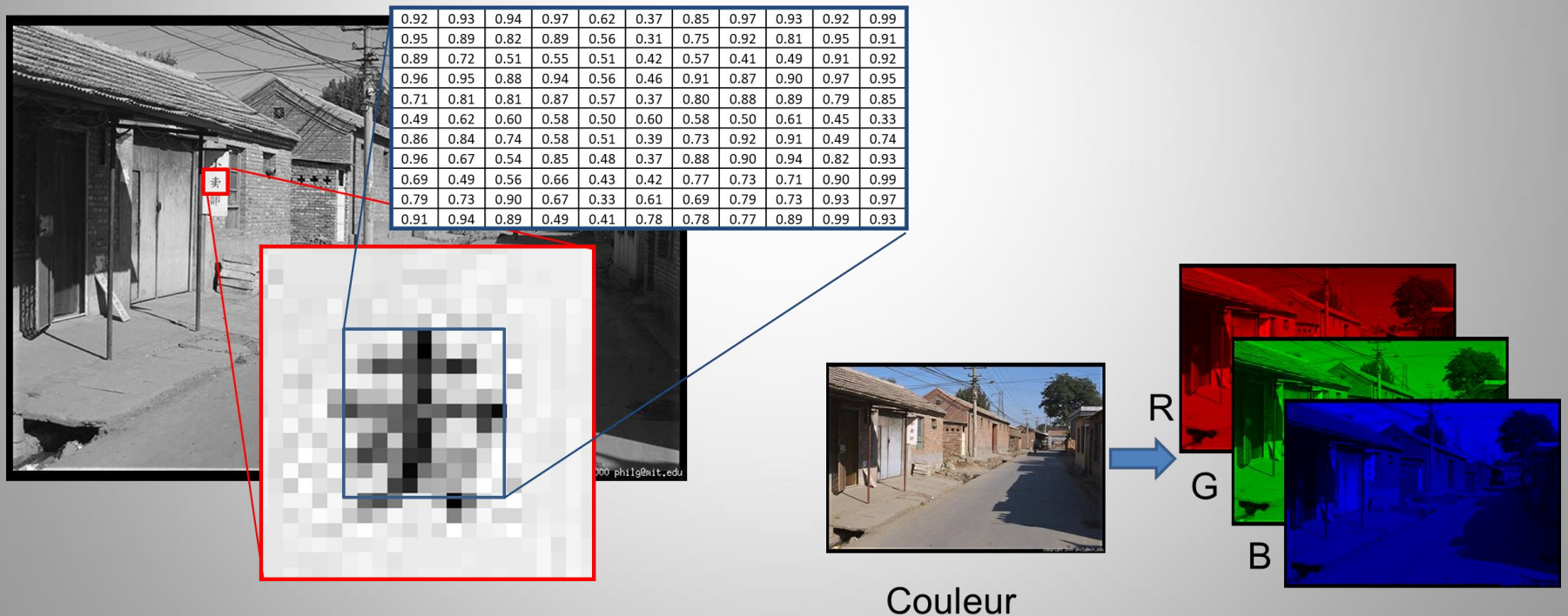


Quelques termes

- **Marqueur** - utilisé pour spécifier où et quelle information ou contenu doit être placé (spécifiques ou image)
- **Primitives naturelles** – points/parties d'un objet visualisé
- **Detecteur** – utilisé pour rechercher dans les images les points spécifiques répétitifs
- **Descripteur** – utilisé pour caractériser les points ou région à partir de l'image. Ils sont utilisés dans la mise en correspondance
- **Canal** – association d'un marqueur à l'objet synthétique à afficher

Rappel du cours précédent

- Image = tableau de valeurs
- Image de couleur = 3 images: Rouge Vert et Bleu (RGB)



Detection et Appariement

- Plusieurs méthodes existent pour décrire, détecter, et appairer les images
- Pixels, points, segments, régions, et droites des images peuvent être utilisées
- Quatre étapes sont nécessaires dans la détection et l'appariement des primitives
 - Détection de primitives
 - Description des primitives
 - Appariement des primitives
 - Tracking de primitives

Exemple de Marqueur image



- Pour faire de la RA, il va falloir
 - Retrouver l'image,
 - la délimiter
 - Dans toutes les conditions (proche, loin, oblique)

Exemple de Marqueur image

- Concrètement il va falloir
 - Avoir un moyen pour décrire l'image de référence



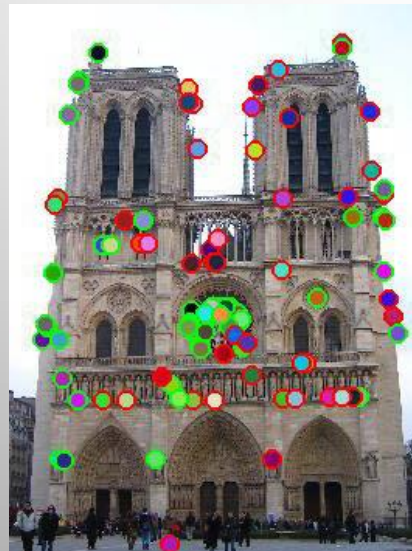
- Avoir un moyen de retrouver
- De le différencier des autres images

-> Vision par ordinateur



Qu'est-ce qu'une primitive

- Une primitive c'est:
- Un élément spécifique de l'image
- Pixels/Point/coin unique de l'image
- Utilisé pour représenter/simplifier l'information contenue dans l'image



Points



Qu'est-ce qu'une primitive

Ca peut être aussi



Segments

Régions

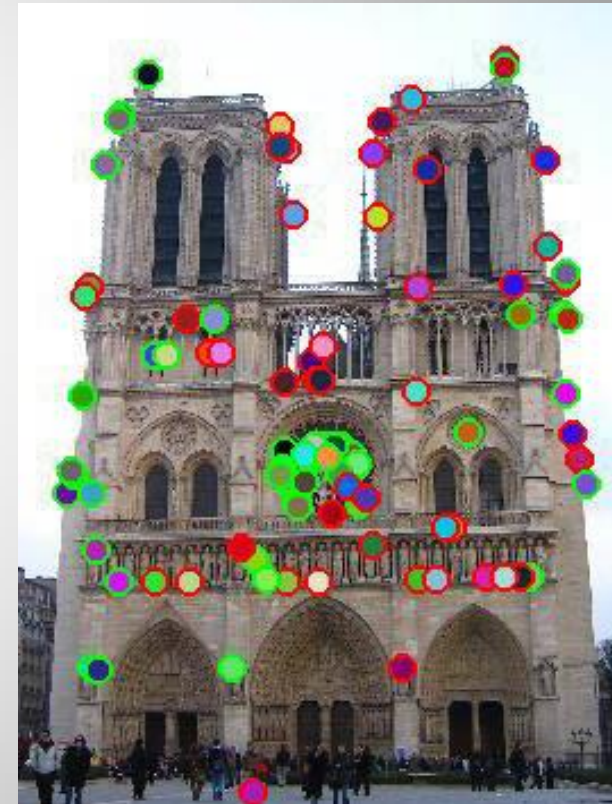


Contours



Détecteur de primitive

- Il va extraire/sélectionner les primitives de l'image
- Critères de qualité:
 - Caractérisables: distinctif, particularité, reconnaissable, précision
 - Répétabilité et invariance: échelle, rotation, illumination, point de vue, bruit



Détecteur de primitive

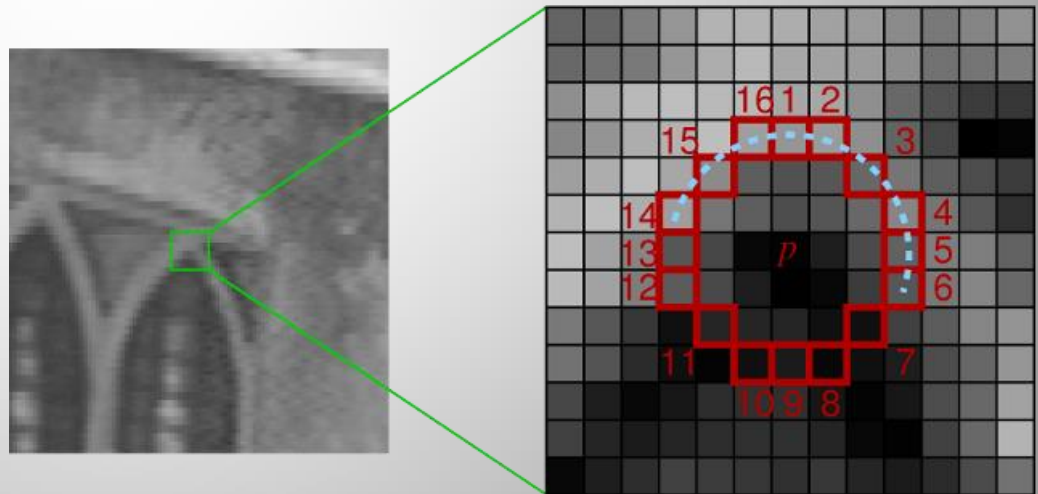


Détection de coins

FAST: Features from Accelerated Segment Test

<http://www.edwardrosten.com/work/fast.html>

- Cercle Bresenham 16 pixels autour du point analysé
- On détecte un coin en p si
l'intensité de N pixels
est $>$ ou $<$ de $X\%$ à I_p
- Rapide et robuste



Descripteur de points

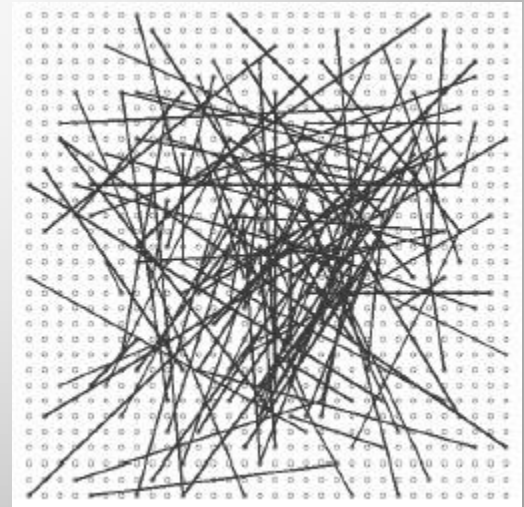
- Description du point à partir de l'image (locale)
- Utilisé pour l'appariement
 - Stockage des descriptions des marqueurs image
 - Comparer avec les primitives de l'image courante
- Critères de qualité:
 - Discriminant
 - Invariant : échelle, rotation, illumination, point de vue, bruit
 - Rapide et empreinte mémoire faible

Descripteur de points

BRIEF : Binary robust independent elementary features

<http://cvlab.epfl.ch/research/detect/brief>

- Vecteur de N paires de points sur un patch
- Comparaison pour chaque paire
 - Si $I_1 < I_2$ alors $c=1$
 - Sinon $c=0$
- Descripteur=100101001...
- Rapide et robuste

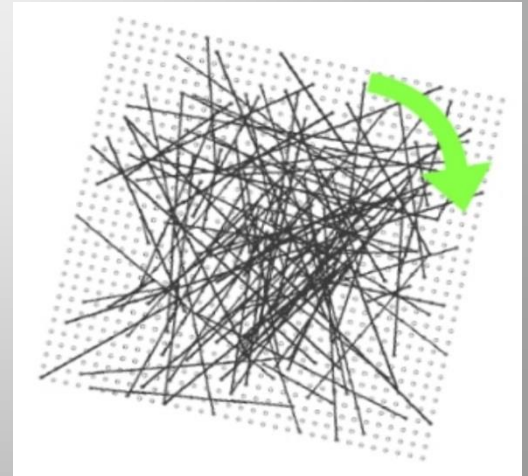
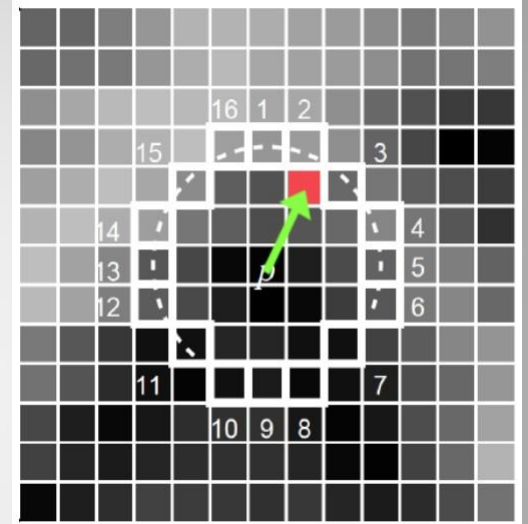


Descripteur de points

ORB (Oriented FAST and Rotated BRIEF)

http://docs.opencv.org/.../py_feature2d/py_orb/py_orb.html

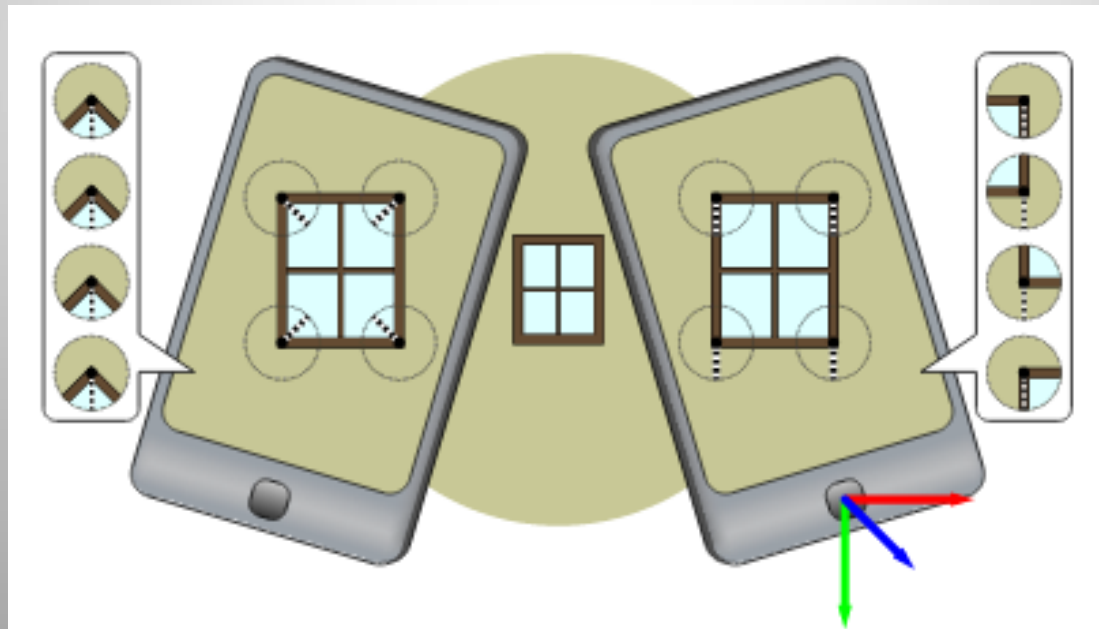
- Prise en compte rotation pour robustesse
- Direction=pixel avec variation la plus forte
- Rotated BRIEF pour aligner les descripteurs lors du matching



Descripteur de points

Autre exemple: GAFD Gravity Aligned Feature Descriptors

- Utilisé par Metaio (Apple)
- Utilise les capteur inertiel pour avoir des descripteurs alignés avec la gravité



Reconnaissance par matching

Appariement des coins

- Brute force matching, on teste toutes les paires
- Similarité= Distance de Hamming (nombre de bits différents)

A = 1 0 1 1 0 0 1 0 0 1 0 0

B = 1 0 0 1 0 0 0 0 1 1 1 1

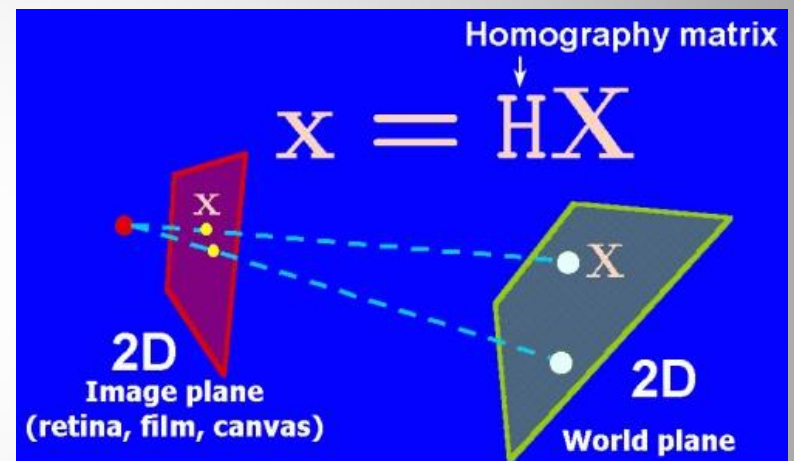
Distance de Hamming = 3

- Si on a un nombre de coins appariées suffisants, l'objet est retrouvé

Relocalisation 2D du pattern

Calcul de l'homographie du plan

- Système d'équation linéaire
- Estimation robuste (RANSAC)
- Filtrage des outliers
- Décomposition en VP

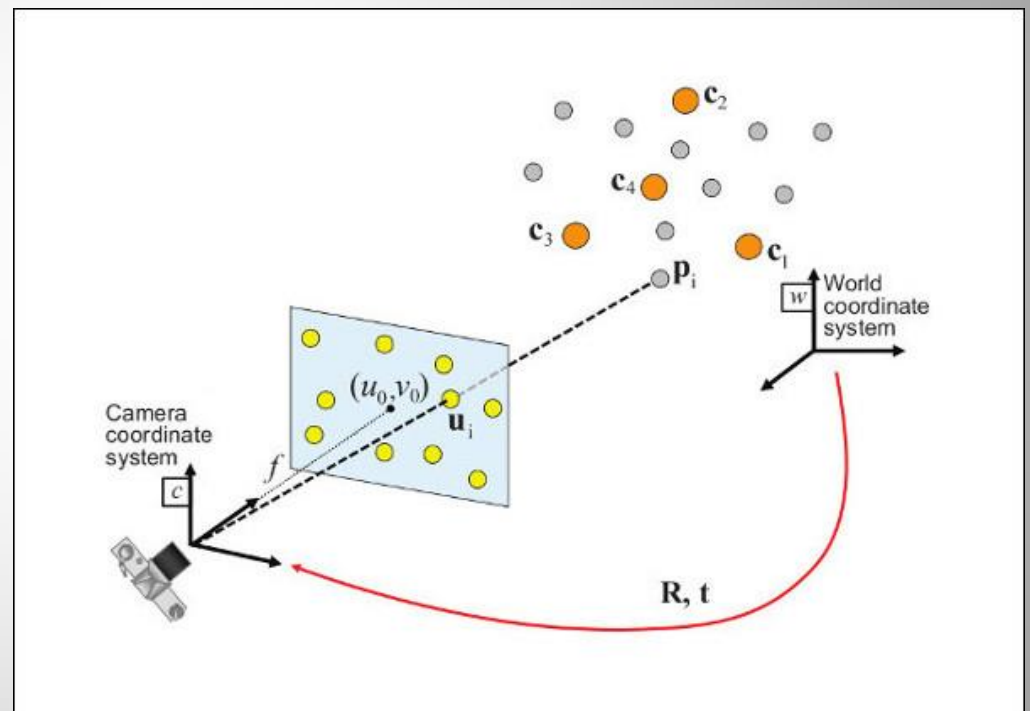


$$\lambda \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \underbrace{\begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix}}_{\text{homography } H} \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix}.$$

Cacul de la Pose 3D

Calcul de la pose de la caméra par rapport à un objet 3D

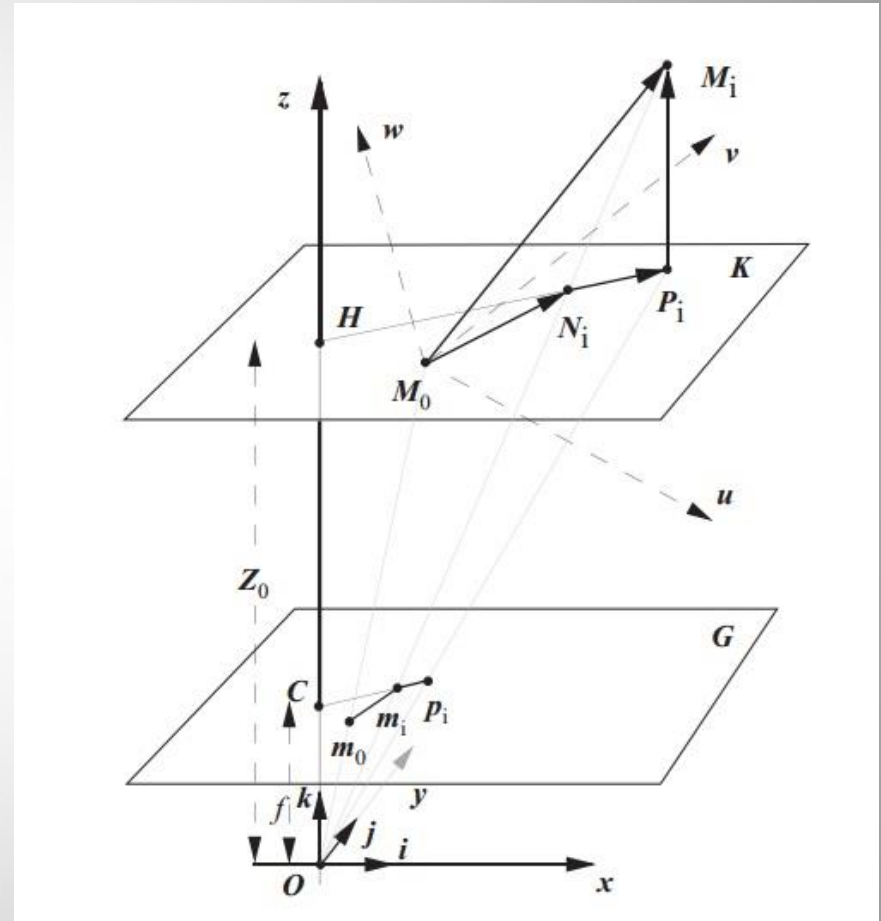
- General case:
 - 6DoF
 - Projection model
- Simplification
 - Calibration connue
 - Perspective-n-Point
 - Projection ortho
 - POSIT



POSIT

POSIT: **P**ose from **O**rthography and **S**caling with **I**terations

- Algorithme itératif pour résoudre PnP non coplanaires
- 4 points coplanaires:
Coplanar POSIT



More on Pose 3D

Calcul de la pose de la caméra par rapport à un objet 3D

- POSIT: [original publications](#), [3D pose estimation](#)
- [Real Time pose estimation](#) : OpenCV tutorial, C++
- [Eric Marchand](#): Article Complet Pose 3D AR
- [Caméra calibration](#): OpenCV tutorial, C++
- [posest](#): C++ opensource
- [Minimal problems](#) in Computer Vision: many links
- Moving camera = Kalman/SLAM

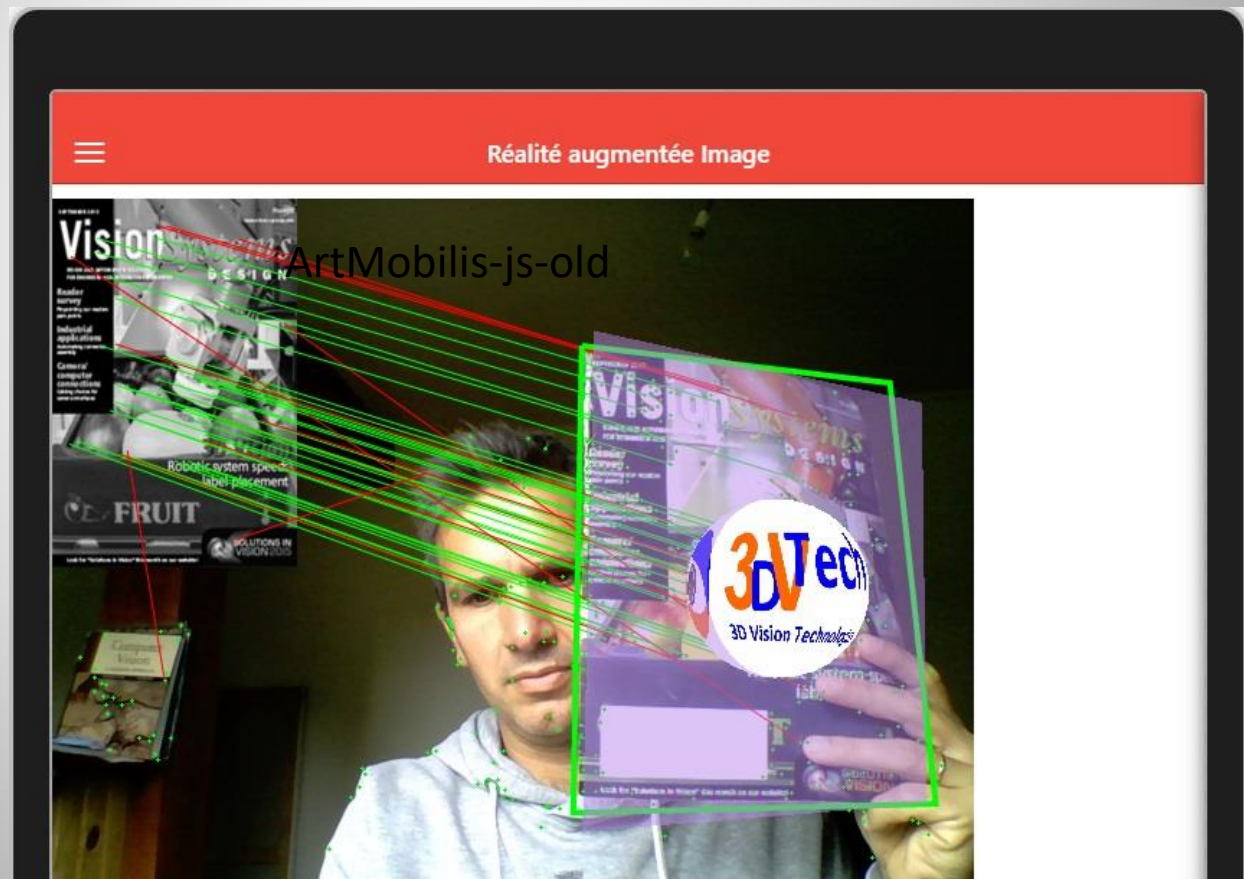
Objectif de ArtMobilis

Un parcours urbain en réalité augmentée

- Géolocalisation des points d'intérêts
- Tracking de la localisation des contenus augmentés
- Support mobile (android, IOS, tablettes)
- OpenSource: <https://github.com/artmobilis/>
- LabMobilis:
 - Implémentation orientée Web pour adaptabilité
 - Application HTML5, CSS3 et JavaScript

Prototype développé

- Demo ArtMobilis-js-old
- Code



Autres outils

AR fondation

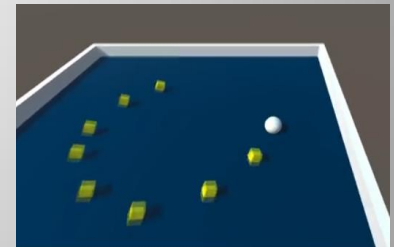
- <https://github.com/Unity-Technologies/arfoundation-samples>

Wikitude

- <https://www.wikitude.com/>

Exercices précédents

- ImageTarget
- Star wars
 - Ground plane, Déplacer et Animer un objet
 - Utiliser Ground Plane Detection (Vuforia) et ajouter objet
 - Ajouter un Canvas + image + fixer orientation smartphone
 - Bouton pour lancer missile
 - Système de particule pour fumée
- Roll a ball
 - Déplacer et Animer un objet
 - Déplacer caméra avec objet
 - Détecter collision d'objets
 - Contrôle par smartphone ([Gyroscope.attitude](https://github.com/vestri/CoursAR))
 - Score et construction du jeu



Matériel: <https://github.com/vestri/CoursAR>

Projet final cours AR

- Objectifs:
 - 1 projet chacun avec AR inside
 - Outil que vous voulez: Unity, Vuforia, JS, Arcore, Arkit...
 - Présentation le dernier cours
- Planning
 - Trouver un sujet/idée en RA pour la semaine prochaine
 - Unity/vuforia cette semaine, JavaScript semaine prochaine

Pour la prochaine fois

- **Finir votre projet**
- **2 à 5 slides de présentation**
- **1 démo sur Smartphone ou PC**
- **Ayez tout le nécessaire pour fbon fonctionnement: cables, adaptateur hdmi, impression...**

Plus d'infos

- **Réalité Augmentée:**
 - RAPRO: <http://www.augmented-reality.fr/>
 - SDK liste: [Social Compare-AR-Sdk](#)
 - Lunettes RA: [Social Compare-AR-lunettes](#)
- **Projet**
 - <https://github.com/artmobilis/>
 - vestri@3DVTech.com